michael kearns + aaron roth

the ethical algorithm

the science of socially aware algorithm design

MICHAEL KEARNS
AND
AARON ROTH

THE ETHICAL ALGORITHM

The Science of Socially Aware Algorithm Design

OXFORD
UNIVERSITY PRESS

OXFORD
UNIVERSITY PRESS

Oxford University Press is a department of the University of Oxford.
It furthers the University's objective of excellence in research, scholarship,
and education by publishing worldwide. Oxford is a registered trade mark of
Oxford University Press in the UK and certain other countries.

Published in the United States of America by Oxford University Press
198 Madison Avenue, New York, NY 10016, United States of America.

Library of Congress Cataloging-in-Publication Data
Names: Kearns, Michael, 1971– author. | Roth, Aaron (Writer on technology), author.
Title: The ethical algorithm : the science of socially aware algorithm design /
Michael Kearns and Aaron Roth.
Description: New York : Oxford University Press, 2019. |
Includes bibliographical references and index. |
Identifiers: LCCN 2019025725 |
ISBN 9780190948207 (hardback) | ISBN 9780190948221 (epub)
Subjects: LCSH: Information technology—Economic aspects. |
Technological innovations—Moral and ethical aspects.
Classification: LCC HC79.I55 K43 2019 | DDC 174/.90051—dc23
LC record available at https://lccn.loc.gov/2019025725

1 3 5 7 9 8 6 4 2
Printed by Sheridan Books, Inc.
United States of America

CONTENTS

THE ETHICAL ALGORITHM

INTRODUCTION

ALGORITHMIC ANXIETY

We are allegedly living in a golden age of data. For practically any question about people or society that you might be curious about, there are colossal datasets that can be mined and analyzed to provide answers with statistical certainty. How do the behaviors of your friends influence what you watch on TV, or how you vote? These questions can be answered with Facebook data, which records the social network activity of billions of people worldwide. Are people who exercise frequently less likely to habitually check their email? For anyone who uses an Apple Watch, or an Android phone together with the Google Fit app, the data can tell us. And if you are a retailer who wants to better target your products to your customers by knowing where and how they spend their days and nights, there are dozens of companies vying to sell you this data.

Which all brings us to a conundrum. The insights we can get from this unprecedented access to data can be a great thing: we can get new understanding about how our society works, and improve public health, municipal services, and consumer products. But as individuals, we aren't just the recipients of the fruits of this data analysis: we *are* the data, and it is being used to make decisions *about us*—sometimes very consequential decisions.

In December 2018, the *New York Times* obtained a commercial dataset containing location information collected from phone apps whose nominal purpose is to provide mundane things like weather reports and restaurant recommendations. Such datasets contain precise locations for hundreds of millions of individuals, each updated hundreds (or even thousands) of times a day. Commercial buyers of such data will generally be interested in aggregate information—for example, a hedge fund might be interested in tracking the number of people who shop at a particular chain of retail outlets in order to predict quarterly revenues. But the data is recorded by individual phones. It is superficially anonymous, without names attached—but there is only so much anonymity you can promise when recording a person's every move.

For example, from this data the *New York Times* was able to identify a forty-six-year-old math teacher named Lisa Magrin. She was the only person who made the daily commute from her home in upstate New York to the middle school where she works, fourteen miles away. And once someone's identity is uncovered in this way, it's possible to learn a lot more about them. The *Times* followed Lisa's data trail to Weight Watchers, to a dermatologist's office, and to her ex-boyfriend's home. She found this disturbing and told the *Times* why: "It's the thought of people finding out those intimate details that you don't want people to know." Just a couple of decades ago, this level of intrusive surveillance would have required a private investigator or a

government agency; now it is simply the by-product of widely available commercial datasets.

Clearly, we have entered a brave new world.

And it's not only privacy that has become a concern as data gathering and analysis proliferate. Because algorithms—those little bits of machine code that increasingly mediate our behavior via our phones and the Internet—aren't simply analyzing the data that we generate with our every move. They are also being used to actively make decisions that affect our lives. When you apply for a credit card, your application may never be examined by a human being. Instead, an algorithm pulling in data about you (and perhaps also about people "like you") from many different sources might automatically approve or deny your request. Though there are benefits to knowing instantaneously whether your request is approved, rather than waiting five to ten business days, this should give us a moment of pause. In many states, algorithms based on what is called machine learning are also used to inform bail, parole, and criminal sentencing decisions. Algorithms are used to deploy police officers across cities. They are being used to make decisions in all sorts of domains that have direct and real impact on people's lives. All this raises questions not only of privacy but also of fairness, as well as a variety of other basic social values including safety, transparency, accountability, and even morality.

So if we are going to continue to generate and use huge datasets to automate important decisions (a trend whose reversal seems about as plausible as our returning to an agrarian society), we have to think seriously about some weighty topics. These include limits on the use of data and algorithms, and the corresponding laws, regulations, and organizations that would determine and enforce those limits. But we must also think seriously about addressing the concerns scientifically—about what it might mean to encode ethical principles directly into the design of the algorithms that are increasingly

woven into our daily lives. This book is about the emerging science of ethical algorithm design, which tries to do exactly that.

SORTING THROUGH ALGORITHMS

But first, what is an algorithm anyway? At its most fundamental level, an algorithm is nothing more than a very precisely specified series of instructions for performing some concrete task. The simplest algorithms—the ones we teach to our first-year computer science students—do very basic but often important things, such as sorting a list of numbers from smallest to largest. Imagine you are confronted with a row of a billion notecards, each of which has an arbitrary number written on it. Your goal is to rearrange the notecards so that the numbers are in ascending order—or, more precisely, to specify an algorithm for doing so. This means that each step of the process you describe must be unambiguous, and that the process must always terminate with the notecards arranged in ascending order, regardless of the numbers and their initial arrangement.

Here is one way to start. Scan through the initial arrangement from left to right to find the smallest of the numbers (you're allowed to use a pencil and paper as storage or memory to help you), which perhaps is written on the 65,704th notecard. Then swap that notecard with the leftmost notecard. Now the smallest number comes first in the list, as desired. Next, rescan the cards starting second from left to find the second-smallest number, and swap its notecard with the second from left. Continue in this fashion until you've completely sorted the list. This is an algorithm—it's precisely specified, and it always works.

It's also a "bad" algorithm, in the sense that there are considerably faster algorithms for the same problem. If we think about it for a moment, we see that this algorithm will scan through the list of a billion numbers a billion times—first from leftmost to rightmost, then from second from left to rightmost, then from third from left to

rightmost, and so on—each time placing just one more number in its proper position. In the language of algorithms, this would be called a quadratic time algorithm, because if the length of the list is *n*, the number of steps or "running time" required by the algorithm would be proportional to the square of *n*. And if *n* is in the billions—as it would be, for example, if we wanted to sort Facebook users by their monthly usage time—it would be infeasibly slow for even the fastest computers. Fortunately for Facebook (and the rest of us), there are algorithms whose running time is much closer to *n* than to its square. Such algorithms are fast enough for even the largest real-world sorting problems.

One of the interesting aspects of algorithm design is that even for fundamental problems such as sorting, there can be multiple alternative algorithms with different strengths and weaknesses, depending on what our concerns are. For example, the extensive Wikipedia page on sorting lists forty-three different algorithms, with names like Quicksort, Heapsort, Bubblesort, and Pigeonhole Sort. Some are faster when we can assume the initial list is in a random order (as opposed to being in reverse sorted order, for example). Some require less memory than others, at the expense of being slower. Some excel when we can assume that each number in the list is unique (as with social security numbers).

So even within the constraint of developing a precise recipe for a precise computational task, there may be choices and trade-offs to confront. As the previous paragraph suggests, computer science has traditionally focused on algorithmic trade-offs related to what we might consider performance metrics, including computational speed, the amount of memory required, or the amount of communication required between algorithms running on separate computers. But this book, and the emerging research it describes, is about an entirely new dimension in algorithm design: the explicit consideration of social values such as privacy and fairness.

MAN VERSUS MACHINE (LEARNING)

Algorithms such as the sorting algorithm we describe above are typically coded by the scientists and engineers who design them: every step of the procedure is explicitly specified by its human designers, and written down in a general-purpose programming language such as Python or C++. But not all algorithms are like this. More complicated algorithms—the type that we categorize as machine learning algorithms—are automatically derived from data. A human being might hand-code the process (or meta-algorithm) by which the final algorithm—sometimes called a model—is derived from the data, but she doesn't directly design the model itself.

In traditional algorithm design, while the output might be useful (like a sorted list of Facebook usage times, which could help in analyzing the demographic properties of the most engaged users), that output is not itself another algorithm that can be directly applied to further data. In contrast, in machine learning, that's the entire point. For example, think about taking a database of high school information about previously admitted college students, some of whom graduated from college and some of whom did not, and using it to derive a model predicting the likelihood of graduation for future applicants. Rather than trying to directly specify an algorithm for making these predictions—which could be quite difficult and subtle—we write a meta-algorithm that uses the historical data to *derive* our model or prediction algorithm. Machine learning is sometimes considered a form of "self-programming," since it's primarily the data that determines the detailed form of the learned model.

This data-driven process is how we get algorithms for more human-like tasks, such as face recognition, language translation, and lots of other prediction problems that we'll talk about in this book. Indeed, with the aforementioned explosion of consumer data enabled by the Internet, the machine learning approach to algorithm design is now

much more the rule than the exception. But the less directly involved humans are with the final algorithm or model, the less aware they may be of the unintended ethical, moral, and other side effects of those models, which are the focus of this book.

HOW THINGS CAN GO WRONG

The reader might be excused for some skepticism about imparting moral character to an algorithm. An algorithm, after all, is just a human artifact, like a hammer, and who would entertain the idea of an ethical hammer? Of course, a hammer might be put to an unethical use—as an instrument of violence, for example—but this can't be said to be the hammer's fault. Anything ethical about the use or misuse of a hammer can be attributed to the human being who wields it.

But algorithms—especially models derived directly from data via machine learning—are different. They are different both because we allow them a significant amount of agency to make decisions without human intervention and because they are often so complex and opaque that even their designers cannot anticipate how they will behave in many situations. Unlike a hammer, which is designed to do only one thing exceptionally well, algorithms can be tremendously general-purpose, closer to the human mind in their flexibility of purpose than to something you'd find in a carpenter's toolbox. And unlike with a hammer, it is usually not so easy to blame a particular misdeed of an algorithm directly on the person who designed or deployed it. In this book, we will see many instances in which algorithms leak sensitive personal information or discriminate against one demographic or another. But how exactly do these things happen? Are violations of privacy and fairness the result of incompetent software developers or, worse yet, the work of evil programmers deliberately coding racism and back doors into their programs?

The answer is a resounding no. The real reasons for algorithmic misbehavior are perhaps even more disturbing than human incompetence or malfeasance (which we are at least more familiar with and have some mechanisms for addressing). Society's most influential algorithms—from Google search and Facebook's News Feed to credit scoring and health risk assessment algorithms—are generally developed by highly trained scientists and engineers who are carefully applying well-understood design principles. The problems actually lie within those very principles, most specifically those of machine learning. It will serve us well later to discuss those principles a bit now.

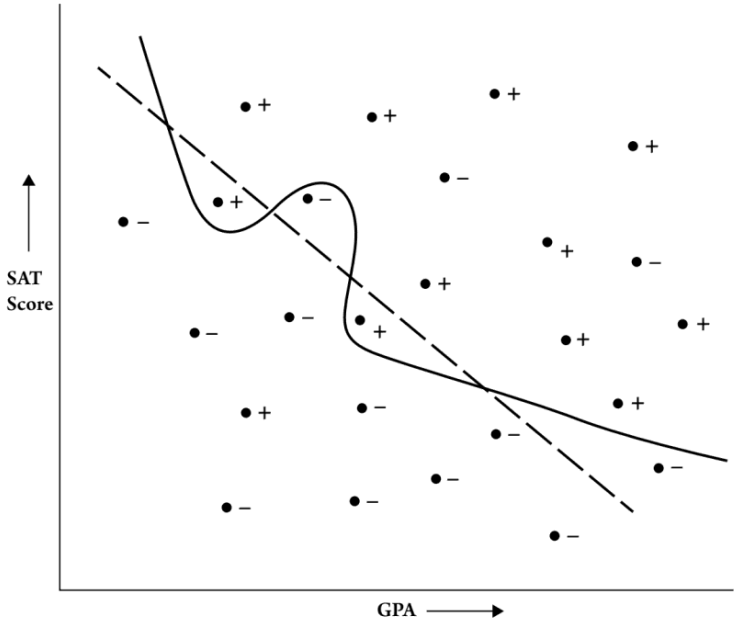

Fig. 1. Building a model to predict collegiate success from high school data. Imagine that each point represents the high school GPA and SAT score of a college student. Points labeled with "+" represent students who successfully graduated from college in four years, while points labeled with "–" represent students who did not. The straight dashed line does an imperfect but pretty good job of separating positives from negatives and could be used to predict success for future high school students. The solid curve makes even fewer errors but is more complicated, potentially leading to unintended side effects.

As we've suggested, many of the algorithms we discuss in this book would more accurately be called *models*. These models, which make the actual decisions of interest, are the result of powerful machine learning (meta-) algorithms being applied to large, complex datasets. A crude but useful sketch of the pipeline is that the data is fed to an algorithm, which then searches a very large space of models for one that provides a good fit to the data. Think of being given a cloud of 100 points on a piece of paper, each labeled either "positive" or "negative," and being asked to draw a curve that does a good but perhaps imperfect job of separating positives from negatives (see Figure 1). The positive and negative points are the data, and you are the algorithm—trying out different curves until you settle on what you think is the best separator. The curve you pick is the model, and it will be used to predict whether future points are positive or negative But now imagine that instead of 100 points, there are 10 million; and instead of the points being on a 2-dimensional sheet of paper, they lie in a 10,000-dimensional space. We can't expect you to act as the algorithm anymore, however smart you might be.

The standard and most widely used meta-algorithms in machine learning are simple, transparent, and principled. In Figure 2 we replicate the high-level description or "pseudocode" from Wikipedia for the famous backpropagation algorithm for neural networks, a powerful class of predictive models. This description is all of eleven lines long, and it is easily taught to undergraduates. The main "forEach" loop is simply repeatedly cycling through the data points (the positive and negative dots on the page) and adjusting the parameters of the model (the curve you were fitting) in an attempt to reduce the number of misclassifications (positive points the model misclassifies as negative, and negative points the model misclassifies as positive). It's doing what you would do, except in a perhaps more systematic, mathematical way, and without any limits on how many data points there are or how complex they are.

```
initialize network weights (often small random values)
do
   forEach training example named ex
      prediction = neural-net-output(network, ex)  // forward pass
      actual = teacher-output(ex)
      compute error (prediction - actual) at the output units
      compute Δw_h for all weights from hidden layer to output layer  // backward pass
      compute Δw_i for all weights from input layer to hidden layer   // backward pass continued
      update network weights // input layer not modified by error estimate
until all examples classified correctly or another stopping criterion satisfied
return the network
```

Fig. 2. Pseudocode for the backpropagation algorithm for neural networks.

So when people talk about the complexity and opaqueness of machine learning, they really don't (or at least shouldn't) mean the actual optimization algorithms, such as backpropagation. These are the algorithms designed by human beings. But the *models* they produce—the outputs of such algorithms—can be complicated and inscrutable, especially when the input data is itself complex and the space of possible models is immense. And this is why the human being deploying the model won't fully understand it. The goal of backpropagation is perfectly understandable: minimize the error on the input data. The opacity of machine learning, and the problems that can arise, are really emergent phenomena that result when straightforward algorithms are allowed to interact with complex data to produce complex predictive models.

For example, it may be that the model that minimizes the overall error in predicting collegiate success, when used to make admissions decisions, happens to falsely reject qualified black applicants more often than qualified white applicants. Why? Because the designer didn't anticipate it. She didn't tell the algorithm to try to equalize the false rejection rates between the two groups, so it didn't. In its standard form, machine learning won't give you anything "for free" you didn't explicitly ask for, and may in fact often give you the opposite of what you wanted. Put another way, the problem is that rich model spaces such as neural networks may contain many "sharp corners" that provide the opportunity to achieve their objective at the expense of other things we didn't explicitly think about, such as privacy or fairness.

The result is that the complicated, automated decision-making that can arise from machine learning has a character of its own, distinct from that of its designer. The designer may have had a good understanding of the algorithm that was used to *find* the decision-making model, but not of the model itself. To make sure that the effects of these models respect the societal norms that we want to maintain, we need to learn how to design these goals directly into our algorithms.

WHO ARE WE?

Before we embark on our journey intertwining technology, society, ethics, and algorithm design, it will be helpful to know a little about who we are and how we came to be interested in these apparently disparate topics. Our backgrounds will in turn illuminate what this book is and is not intended to be about, and what we are more and less qualified to opine upon.

For starters, we are both career researchers in the field known as theoretical computer science. As the name somewhat generically suggests, this is the branch of computer science with particular interest in formal, mathematical models of computation. We deliberately say "computation" and not "computers," because for the purposes of this book (and perhaps even generally), the most important thing to know about theoretical computer science is that it views computation as a ubiquitous phenomenon, not one that is limited to technological artifacts. The scientific justification for this view originates with the staggeringly influential work of Alan Turing (the first theoretical computer scientist) in the 1930s, who demonstrated the universality of computational principles with his mathematical model now known as the Turing machine. Many trained in theoretical computer science, ourselves included, view the field and its tools not simply as another scientific discipline but as a way of seeing and understanding the world around us—perhaps much as those trained in theoretical physics in an earlier era saw their own field.

So a theoretical computer scientist sees computation taking place everywhere—certainly in computers, but also in nature (in genetics, evolution, quantum mechanics, and neuroscience), in society (in markets and other systems of collective behavior), and beyond. These areas all embody computation in the general sense that Turing envisioned. Certainly the physical mechanisms and details differ—computation in genetics, for example, involves DNA and RNA instead of the circuits and wires of traditional electronic computers, and is less precise—but we can still extract valuable insights by treating such varied systems as computational devices.

This worldview is actually shared by many computer scientists, not only the theoretical ones. The distinguishing feature of theoretical computer science is the desire to formulate mathematically precise models of computational phenomena and to explore their algorithmic consequences. A machine learning practitioner might develop or take an algorithm like backpropagation for neural networks, which we discussed earlier, and apply it to real data to see how well it performs. Doing so doesn't really require the practitioner to precisely specify what "learning" means or doesn't mean, or what computational difficulties it might present generally. She can simply see whether the algorithm works well for the specific data or task at hand.

In contrast, a theoretical computer scientist would be inclined to approach machine learning by first giving a precise definition (or perhaps multiple variations on an underlying definition) of "learning," and then to systematically explore what can and cannot be achieved algorithmically under this definition. With tongue only slightly in cheek, we can view the typical practitioner as following Nike's "Just Do It" mantra, while the theorist follows the "Just Define and Study It" mantra. When put this way, people might naturally wonder what the practical value of theoretical computer science is, and it's true that in many scientific disciplines theory often lags behind practice. But we would argue (and suspect many of our colleagues would

agree) that the theoretical approach is essential when the right definition is far from clear, and when getting it right matters a lot, as with concepts such as "privacy" and "fairness."

Writing down precise definitions that capture the essence of critical and very human ideas without becoming overly complex is something of an art form, and it is inevitable that in many settings, simplifications—sometimes painful ones—are necessary. We will see this tension arise repeatedly throughout this book. But we should keep in mind that this tension is not an artifact of the theoretical approach per se; instead, it reflects the inherent difficulty of being precise about concepts that previously have been left vague, such as "fairness." We believe that the only way to make algorithms better behaved is to begin by specifying what our goals for them might be in the first place.

Our training notwithstanding, our research and interest in the topics described here have not been formulated in a vacuum of abstraction and mathematics. We've both always been interested in applying that approach to problems in machine learning and artificial intelligence. We are also neither adverse to nor inexperienced in experimental, data-driven work in machine learning—often as a test of the practicality and limitations of our theories, but not always. And it was the very trends we describe in these pages—the explosive growth of consumer data enabled by the Internet, and the accompanying rise in machine learning for automated decision-making—that made us and our colleagues aware of and concerned about the potential collateral damage.

We have spent much of the last decade researching the topics that we cover in this book, and engaging with a variety of stakeholders. We've spent many hours talking to lawyers, regulators, economists, criminologists, social scientists, technology industry professionals, and many others about the issues raised in these pages. We've provided testimony and input to congressional committees, corporations, and government agencies on algorithmic privacy and fairness. And between us we have extensive, hands-on professional experience in areas

including quantitative trading and finance; legal, regulatory, and algorithmic consulting; and technology investing and start-ups—all of which are beginning to confront the social issues that are our themes here.

We are, in short, modern computer scientists. We also know what we are not, and should not pretend to be. We are not lawyers or regulators. We are not judges, police officers, or social workers. We are not on the front lines, directly seeing and helping the people who suffer harms from privacy or fairness violations by algorithms. We are not social activists with a deep, firsthand sense of the history and problems of discrimination and other forms of injustice.

Because of this, we'll tend to say relatively little on important matters where such expertise is essential, such as designing better laws or policies, proposing how to improve social agencies to reduce unfairness in the first place, or opining on whether and how to stem labor displacement resulting from technology. It's not that we don't care about these topics or have opinions on them; we do. But doing justice to them would have fundamentally altered the nature of the book we wanted to write, which is about scientific approaches and solutions to what we might call socio-algorithmic problems.

So we will stick to what we know well and have thought hard about: how to design better algorithms.

WHAT THIS BOOK IS (NOT) ABOUT

It doesn't take much searching to discover many recent books, news stories, and scientific articles documenting cases in which algorithms have caused harms to specific people, and often to large groups of people. For instance, controlled online experiments have demonstrated racial, gender, political, and other types of bias in Google search results, Facebook advertising, and other Internet services. An explosive controversy over racial discrimination in the predictive models used in

criminal sentencing decisions has recently consumed statisticians, criminologists, and legal scholars. In the domain of data privacy, there have been many cases in which sensitive information about specific people—including medical records, web search behavior, and financial data—has been inferred by "de-anonymizing" data that was allegedly made "anonymous" by algorithmic processing (as in the aforementioned *New York Times* case about location data and Lisa Magrin). Turbocharged by algorithmic data analysis tools that make it faster and more efficient to search for correlations in data, there has even been a rash of reported scientific findings that turn out not to be true, costing both dollars and lives. It has become very clear that modern algorithms may routinely trample on some of our most cherished social values.

So the problems are becoming obvious. What about the solutions? Much of the discussion to date has focused on what we might consider to be "traditional" solutions, such as new laws and regulations focused on algorithms, data and machine learning. The European Union's General Data Protection Regulation is a sweeping set of laws designed to limit algorithmic violations of privacy and to enforce still-vague social values such as "accountability" and "interpretability" on algorithmic behavior. Legal scholars are immersed in discussions of the ways existing laws do or do not apply to previously human-centric arenas that are increasingly dominated by algorithms, such as Title VII's prohibition of discrimination in employment decisions in the United States. The tech industry itself is starting to develop self-regulatory initiatives of various types, such as the Partnership on AI to Benefit People and Society. Government organizations and regulatory agencies are struggling to figure out how the algorithmic landscape affects their missions; the US State Department even held a workshop on the role and influence of AI in foreign policy.

There are important conversations going on, even as we write, about the proper role of data collection in our society: maybe there are

certain things that just shouldn't be done, because the long-term social consequences aren't worth the gains. It might be beside the point whether or not facial recognition algorithms have higher error rates on black people compared to white people; maybe we shouldn't be engaging in large-scale facial recognition at all, simply because it leads us closer to being a surveillance state. All of this activity and debate is healthy, important, and essential and has been written about at length by others.

So rather than covering familiar ground, our book instead dives headfirst into the emerging science of designing social constraints directly into algorithms, and the consequences and trade-offs that emerge. Despite our focus on concrete technical solutions in the rest of this book, we are not under the misapprehension that technology alone can solve complicated social problems. But neither can decisions about our society be made in a vacuum. To make informed decisions, we need to be able to understand the consequences of deploying certain kinds of algorithms, and the costs associated with constraining them in various ways. And that is what this book is about.

At this juncture you'd be forgiven for feeling a tad queasy about a book on ethical algorithms written by theoretical computer scientists. The same people who brought you the disease have now proposed the cure—and it's more algorithms! But we indeed believe that curtailing algorithmic misbehavior will itself require more and better algorithms—algorithms that can assist regulators, watchdog groups, and other human organizations to monitor and measure the undesirable and unintended effects of machine learning and related technologies. It will also require versions of those technologies that are more "socially aware" and thus better-behaved in the first place. This book is about the new science underlying algorithms that internalize precise definitions of things such as fairness and privacy—specified by humans—and make sure they are obeyed. Instead of people regulating and monitoring algorithms from the outside, the idea is to fix

them from the inside. To use one of the nascent field's acronyms, the topics we examine here are about the FATE—fairness, accuracy, transparency, and ethics—of algorithm design.

There is no getting around the fact that the people who have developed a particular branch of science or technology in the first place are almost always the ones most deeply familiar with its limitations, drawbacks, and dangers—and possibly how to correct or reduce them. So it's essential that the scientific and research communities who work on machine learning be engaged and centrally involved in the ethical debates around algorithmic decision-making. Consider the Manhattan Project to develop the atomic bomb during World War II, for instance, and the subsequent efforts of many of its scientists to curtail the use of their own invention for years after. Of course, the loss of human life from algorithms has, mercifully, been far less severe (at least so far) than from the use of nuclear weapons, but the harms are more diffuse and harder to detect. Whatever one believes about the ultimate role that algorithms should play in our society, the idea that their designers should inform that role is fundamentally sound.

The efforts described in these pages in no way propose that algorithms themselves should decide, or even be used to decide, the social values they will enforce or monitor. Definitions of fairness, privacy, transparency, interpretability, and morality should remain firmly in the human domain—which is one of the reasons that the endeavor we describe must ultimately be a collaboration between scientists, engineers, lawyers, regulators, philosophers, social workers, and concerned citizens. But once a social norm such as privacy is given a precise, quantitative definition, it can be "explained" to an algorithm that then makes sure to obey it.

Of course, one of the greatest challenges here is in the development of quantitative definitions of social values that many of us can agree on. And we'll see that this challenge has been met relatively well

so far (if inevitably imperfectly) in areas such as privacy, is making good but murkier progress in areas such as fairness, and has a much longer way to go for values such as interpretability or morality. But despite the difficulties, we argue that the effort to be exceedingly precise by what we mean when we use words such as *privacy* and *fairness* has great merit in its own right—both because it is necessary in the algorithmic era and because doing so often reveals hidden subtleties, flaws, and trade-offs in our own intuitions about these concepts.

A BRIEF PREVIEW

In this book we will see how it is possible to expand the principles on which machine learning is based to demand that they incorporate—in a quantitative, measurable, verifiable manner—many of the ethical values we care about as individuals and as a society.

Of course, the first challenge in asking an algorithm to be fair or private is agreeing on what those words should mean in the first place—and not in the way a lawyer or philosopher might describe them, but in so precise a manner that they can be "explained" to a machine. This will turn out to be both nontrivial and revealing, since many of the first definitions we might consider turn out to have serious flaws. In other cases we will see there may be several intuitive definitions that are actually in conflict with each other.

But once we've settled on our definitions, we can try to internalize them in the machine learning pipeline, encoding them into our algorithms. But how? Machine learning already has a "goal," which is to maximize predictive accuracy. How do we introduce new goals such as fairness and privacy into the code without "confusing" the algorithm? The short answer is that we view these new goals as constraints on the learning process. Instead of asking for the model that only minimizes error, as we ask for the model that minimizes error *subject to the constraint* that it not violate particular notions of fairness or privacy "too

much." While this might be a harder problem computationally, conceptually it is only a slight variation on the original—but a variation with major consequences.

The first major consequence is that we will now have algorithms that are guaranteed to have the particular ethical behaviors we asked for. But the second major consequence is that these guarantees will come at a cost—namely, a cost in the accuracy of the models we learn. If the most accurate model for predicting loan repayment is racially biased, then, by definition, eradicating that bias results in a less accurate model. Such costs may present hard decisions for companies, their regulators, their users, and society at large. How will we feel if more fair and private machine learning results in worse search results from Google, less efficient traffic navigation from Waze, or worse product recommendations from Amazon? What if asking for fairness from criminal sentencing models means more criminals are freed or a greater number of innocent people are incarcerated?

The good news is that the trade-offs between accuracy and good behavior can also be quantified, allowing stakeholders to make informed decisions. In particular, we will see that both accuracy and the social values we consider can be put on sliding scales that are under our control. We'll see concrete examples of this when it comes to privacy in Chapter 1 and fairness in Chapter 2.

Our discussion so far largely refers to the use of machine learning in an isolated, sequential fashion, in which models are built from personal data in order to make predictions or decisions about future individuals. But there are many settings in which there are complex feedback loops between users, the data they generate, the models constructed, and the behavior of those users. Navigation apps use our GPS data to model and predict traffic, which in turn influences the driving routes they suggest to us, which then alters the data used to build the next model. Facebook's News Feed algorithm uses our feedback to build models of the content we want to see, which in turn

influences the articles and posts we read and "like," which again changes the model. The entire system of users, data, and models is perpetually changing and evolving, often in self-interested and strategic ways that we'll examine. We'll even take this view of scientific research itself, as well as its recent "reproducibility crisis." In order to understand such systems, and design them in ways that encourage good social outcomes, we will require some additional science that marries algorithm design with economics and game theory.

What we've just outlined already provides a crude road map to the rest of the book. Chapters 1 and 2 will consider algorithmic privacy and fairness in turn. In our view, these are the areas of ethical algorithms in which the most is known, and where there are relatively mature frameworks and results to discuss. Chapter 3 considers strategic feedback loops between users, data and, algorithms, but it is connected to the previous chapters by its focus on the societal consequences of algorithmic behavior. This leads us to Chapter 4, which focuses on data-driven scientific discovery and its modern pitfalls. Chapter 5 provides some brief thoughts on ethical issues that we consider important but about which we feel there is less to say scientifically so far—issues such as transparency, accountability, and even algorithmic morality. Finally, in the conclusion we briefly distill some lessons learned.

It is important to emphasize at the outset that it will never alone suffice to formalize the social values we care about and design them into our algorithms—it is also essential that such algorithms actually be adopted on a large scale. If platform companies, app developers, and government agencies don't care about privacy or fairness (or if in fact those norms run counter to their objectives), then without encouragement, pressure or coercion they will ignore the types of algorithms we will be describing in these pages. And in our current technological environment, it may often feel like there is indeed a great mismatch between the values of society and the corporations